

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

RAM DISK BOOT OF OPTICAL MEDIA IMAGE

Inventor(s):

Wesley Miller

Charles T. Lenzmeier

Saad Syed

ATTORNEY'S DOCKET NO. MS1-1926US

CLIENT'S DOCKET NO. 306159.01

EV436703015

RAM DISK BOOT OF OPTICAL MEDIA IMAGE

TECHNICAL FIELD

[0001] The described subject matter relates to electronic computing, and more particularly to systems and methods for random access memory (RAM) disk booting using an optical media image.

BACKGROUND

[0002] Booting a computer refers to a process of loading an initial piece of software that starts the computer. Because the computer operating system (OS) is essential for running all other programs, the OS is usually the first piece of software that is loaded. During the boot process, the OS is typically copied from a permanent memory location (e.g., a hard disk) to an impermanent memory location (e.g., random access memory (RAM)). In some specialized computers, such as embedded systems, the OS is copied into a RAM disk.

[0003] A RAM disk is a memory-resident program which mimics a hard disk drive using part of the computer's RAM to store data which can be accessed as files. Unfortunately, RAM disks are not typically available in general purpose computers. Implementing a RAM disk has typically required significant effort by the computer user to retrofit the OS. As a result, traditional RAM disks are not readily realized or sufficiently robust to handle optical media formats.

1 [0004] A RAM disk acts as a virtual drive on the computer. A RAM disk
2 allows one to create directories, copy files to and from memory, and so on. The
3 data however is not written onto a hard disk, but remains purely stored into a
4 particular part of RAM. Because physical disks have mechanical parts that allow
5 for seeking to a particular position on the magnetic storage media and to read and
6 write data, physical disks are relatively slow. In some cases, a RAM disk can read
7 and write the same data thirty to sixty times faster than a hard disk. However, the
8 data stored in RAM is "volatile", meaning that the data disappears when power is
9 removed from the RAM.
10

11 [0005] Unfortunately, traditional RAM disks do not easily allow for
12 representations of software programs embodied on optical media. A primary
13 reason for this is that traditional RAM disks use a cylinder/head/sector format,
14 which is the format traditionally used by disk drives. Thus, for example, a volume
15 accessed by a traditional RAM disk may be a snapshot of an actual system volume
16 on a physical disk and partition in a hard disk drive. Because traditional RAM
17 disks employ a cylinder/head/sector format, and optical media employ a different
18 format (e.g., universal disk format (UDF) or ISO-9660 format), an optical media
19 disk cannot be exactly represented by a traditional RAM disk.
20

21 [0006] The inability to exactly represent optical media using a traditional
22 RAM disk is a substantial drawback, particularly when considering that the most
23 commonly used form factor for booting the OS outside of the hard disk itself is the
24
25

compact disc (CD), and may soon be the digital video disk (DVD), both of which are optical media. Transferring an operating system image from optical media to a traditional RAM disk format and accessing the media from the traditional RAM disk may not be simple and may incur unnecessary costs.

[0007] In addition, some computers may not have an optical media drive, or the optical media drive may be unavailable for booting the OS. In such circumstances, a program may not be exactly represented in a RAM disk if the program is provided on optical media (e.g., CD or DVD). The requirement for a physical CD in such situations can be a burden. As such, traditional RAM disk approaches do not provide the flexibility and robustness often desired by computer users.

SUMMARY

[0008] Implementations are described and claimed herein for booting a computer using a RAM disk based on an optical media format. An optical media image is loaded from a source into RAM and accessed from RAM. The source may be local to the computer or a remote source accessible over a network. Once loaded into RAM, other drives, memory, and resources can be freed for use in activities other than boot-related activities or OS-related activities.

[0009] In some implementations, articles of manufacture are provided as computer program products. One implementation of a computer program product

1 provides a computer program storage medium readable by a computer system and
2 encoding a computer program for booting a computer using a RAM disk image of
3 optical media in an optical media format. Another implementation of a computer
4 program product may be provided in a computer data signal embodied in a carrier
5 wave by a computing system and encoding the computer program for booting a
6 computer using a RAM disk image of optical media in an optical media format.

7 [0010] The computer program product encodes a computer program for
8 executing on a computer system a computer process that loads an optical media
9 image from a source into RAM. The process further includes emulating an optical
10 image from a physical source by the optical image in RAM. Loading the optical
11 image may involve downloading the optical image from a network boot server.

12 [0011] In another implementation, a method includes identifying an optical
13 media image source having an optical media image from which to boot a
14 computer, and loading at least a portion of the optical media image into a RAM
15 from the optical media image source, without changing the optical media image
16 format. Loading may involve creating a RAM disk wherein the portion of the
17 optical media image is in an optical media format. The method may further
18 include accessing a resource from the optical media image from the RAM disk in
19 response to receiving a call from an application to access the resource on a
20 physical optical device.
21
22
23
24
25

1 [0012] In yet another implementation, a system includes a loader loading an
2 optical media image from an optical media image source and onto a RAM disk, the
3 optical media image being formatted in an optical media image format. The system
4 further includes a RAM disk driver accessing the RAM disk according to the
5 optical disk format. The optical media image may include an optical media file
6 system. The RAM disk driver may redirect a request for a resource on an optical
7 media device to a location on the RAM disk containing the resource.
8

9 **BRIEF DESCRIPTION OF THE DRAWINGS**

10
11 [0013] Fig. 1 illustrates an exemplary operating environment for RAM disk
12 booting of an optical media image;

13 [0014] Figs. 2-4 are flow charts illustrating operation flows or algorithms
14 for use in RAM disk booting of an optical media image; and

15 [0015] Fig. 5 is a schematic illustration of an exemplary computing device
16 that can be utilized to implement RAM disk booting of an optical media image.
17

18 **DETAILED DESCRIPTION**

19 **Overview**

20
21 [0016] Briefly, an image loader creates an image of optical media content in
22 RAM where the optical media image is accessible by a RAM disk program. The
23 optical media image is in an optical media format. Because traditional RAM disks
24
25

1 are not designed to access RAM data in an optical media format, optical media
2 images typically have not exactly matched the corresponding optical media
3 content. Fortunately, the RAM disk described herein is able to access the optical
4 media image according to the optical media format. As such, accesses to a
5 physical optical media image source by a program are transparently directed to the
6 RAM disk.

8 **Exemplary System**

9 [0017] Fig. 1 illustrates an exemplary operating environment 100 booting an
10 optical media image 102 via a random access memory (RAM) disk 104 in a
11 computer 106. When the computer 106 powers up, code in a basic input/output
12 system (BIOS) 108 locates and calls code at a boot sector 110. The boot sector
13 110 is typically a sector on a physical memory medium, such as a compact disc
14 (CD), DVD, hard disk or floppy disk. The boot sector 110 loads an image loader
15 112 into RAM 114 to begin the booting process. In certain implementations of the
16 computer 106, the boot sector 110 is not used, but rather the BIOS 108 downloads
17 the image loader 112 from a remote source, such as a pre-boot execution
18 environment (PXE) server.
19
20

21 [0018] The image loader 112 is typically implemented in software, but may
22 include hardware and/or firmware as necessary to carry out an image loading and
23 booting process. The image loader 112 copies optical media content 116 from an
24
25

1 optical media image source 118 to a location in RAM 114 to create the optical
2 media image 102. The image loader 112 communicates with the optical media
3 source 118 via a communication path 120.

4 [0019] The optical media image source 118 may be local with respect to the
5 computer 106 or remote from the computer 106. Thus, the communication path
6 120 may be a data bus, for example, in the local case, or a network, in the remote
7 case. Various exemplary data busses and networks are discussed with respect to
8 the computer system shown in Fig. 5.

9 [0020] The optical media image source 118 is any source of content that can
10 be stored in an optical media format, such as, but not limited to, universal disk
11 format (UDF), UDF-bridge, Joliet, or an International Standards Organization
12 (ISO) 9660 optical media format (and derivatives). Exemplary implementations of
13 the optical media image source 118 include a hard disk drive, an optical disk drive,
14 or a network boot server, such as a server running the WINDOWS REMOTE
15 INSTALLATION SERVICES (RIS) from MICROSOFT CORPORATION. The
16 optical media image source 118 stores the optical media content 116 and makes the
17 optical media content 116 available for booting the computer 106.

18 [0021] The optical media content 116 is content that is formatted in any
19 optical media format. The optical media content 116 can include any type of
20 computer data and/or code. Data and code in the optical media content 116 may or
21 may not be compressed. In one particular implementation, the optical media
22
23
24
25

1 content 116 contains operating system (OS) code, such as the Windows
2 Preinstallation Environment (Windows PE) from MICROSOFT CORPORATION.
3 The OS code includes the OS kernel, software interfaces, and other software and
4 data required for the OS to execute.

5 [0022] In general, the optical media image 102 is a section of random access
6 memory (RAM) that has been copied from another memory or storage location. In
7 the implementation shown in Fig. 1, the optical media image 102 is a copy of the
8 optical media content 116. Thus, the optical media image 102 (also referred to as
9 an optical media image) contains the OS that will be loaded on the computer 106.
10 The optical media image 102 is in an optical media format.

12 [0023] The image loader 112 may also load drivers 122 and/or other data
13 124 required by the optical media image 102 during operation. Drivers 122 can
14 include, by way of example, display drivers, disk drivers, network drivers, printer
15 drivers, speaker drivers, and others. Other data 124 includes any other data, files,
16 and data structures that may be used by the drivers 122, the optical media image
17 102, and/or other application programs 126 executing on the computer 106.

19 [0024] One implementation of the image loader 112 accesses an information
20 file to determine what needs to be loaded in order to boot the computer 106. The
21 information file lists what drivers and other data need to be loaded into RAM 114,
22 as well as the locations from which the drivers and other data can be obtained.
23 The information file also identifies the optical media content 116 to be loaded, as
24

well as the location from which the optical media content file can be obtained.

The image loader 112 copies the optical media content 116 into RAM 114. While copying the optical media content 116, the image loader 112 may decompress the optical media content 116 if it is compressed and requires decompression before it can be loaded.

[0025] After the image loader 112 loads the optical media image 102, the drivers 122, and the other data 124, the image loader sets up a logical optical media partition 128 containing the optical media image 102, the drivers 122, and the other data 124. The image loader 112 assigns a unique drive letter to the optical media partition 128 whereby the optical media image 102, the drivers 122, and the other data 124 can be addressed outside of the context of the drive it may have been loaded from. Beneficially, files and data in the optical media partition 128 can be accessed as a virtual drive.

[0026] An optical media file system (OMFS) driver 130 and the RAM disk driver 132 are launched to provide mechanisms for accessing the optical media partition 128. The OMFS driver 130 includes software and data for managing files in the optical media partition 126. The OMFS driver 130 recognizes the structure of the optical media partition 128 and reads data from (and may write data to, depending on format and implementation) the optical media partition 128 according to the appropriate optical media format. An exemplary implementation of the OMFS system driver 130 is an ISO, ISO-9660 (or derivative) file system.

1 The OMFS driver 130 interacts with the RAM disk driver 132 to perform accesses
2 to the optical media partition 128.

3 [0027] The RAM disk driver 132 includes software and data for performing
4 any accesses to the optical media partition 128. Such software would serve as an
5 abstraction between any typical file system calls made to an optical drive under
6 typical optical media booting conditions, and would redirect these read or write
7 operations to the virtual disk image existing in the optical media partition 128.
8 Thus, for example, the RAM disk driver 132 requests data at certain memory
9 locations within the optical media partition 128 from the OS. The RAM disk
10 driver 132 and the OMFS driver 130 present a virtual drive to an application 126
11 that may access the optical media partition 128. The application 126 requests files
12 through the OMFS driver 130. Although a file request may specify a physical
13 drive letter, the requested file is actually obtained from the optical media partition
14 128. Thus, for example, if the application 126 requests a file from drive 'D:', the
15 OMFS driver 130 and the RAM disk driver 132 will redirect the request to the
16 optical media partition 128 if physical drive 'D:' is represented by the optical
17 media partition 128.
18
19
20

21 Exemplary Operations for RAM disk boot of an Optical Media Image

22 [0028] Described herein are exemplary methods for booting of an optical
23 media image in an optical media format using a RAM disk. The methods described
24
25

1 herein may be embodied as logic instructions on one or more computer-readable
2 medium. When executed on a processor, the logic instructions cause a general
3 purpose computing device to be programmed as a special-purpose machine that
4 implements the described methods. In the following exemplary operations, the
5 components and connections depicted in the figures may be used to implement
6 RAM disk booting of an optical media image in a network environment.

7 [0029] Fig. 2 illustrates a booting operation flow or algorithm 400 for
8 booting an optical media image using a RAM disk when the source of the optical
9 media image is a remote computer running as a remote boot server. Thus, the
10 algorithm 400 may be executed by a client computer that boots its operating
11 system (OS) from optical media content received over a network from a remote
12 computer.
13

14 [0030] In the particular implementation shown in Fig. 2, an initiating
15 operation 202 initiates a booting process via a network card. The initiating
16 operation 202 involves identifying the remote computer from which the OS
17 software can be obtained. The initiating operation 202 contacts the remote
18 computer, and may indicate to the remote computer the OS that the client needs.
19 In response, the remote computer transmits image loader software to the client.
20

21 [0031] A downloading operation 204 downloads the image loader to the
22 client. The image loader includes code for downloading the optical media image,
23 including the OS needed by the client.
24
25

1 [0032] Accordingly, in another downloading operation 206, the boot loader
2 downloads the optical media content, including the OS from the remote computer.
3 The boot loader stores the downloaded content in an optical media image in RAM.
4 After the optical media image is stored in RAM, a disk representation is created
5 for virtually representing a physical disk to the OS and higher level application
6 programs.

7 [0033] A jumping operation 208 jumps to a specified location in the optical
8 media image in RAM to finish the boot process. The jumping operation 208
9 typically jumps to the OS kernel, which controls memory management, disk
10 management, and process and task management.

12 [0034] A particular implementation of an operation for booting from a
13 WINDOWS REMOTE INSTALLATION SERVICE (RIS), from MICROSOFT
14 CORPORATION, is shown next. First, the following directory should be created
15 on the WINDOWS RIS server: \RemoteInstall\Setup\<language>\Images\<OS>.
16 The designation <language> could be "English", for example. The <OS>
17 designation specifies the OS being loaded. By way of example, the OS could be
18 WINPE. In this directory, the following set of files is created:
19

21 \<platform>
22 <bootimage>
23 \templates
24 ntdetect.com
25 ntldr (renamed from setupldr.exe)
winpe.sif
startrom.com

1 <platform> is, for example, "i386".

2 <bootimage> is the boot image file. The <bootimage> can be moved farther
3 down in the directory tree.

4 [0035] In this implementation, the winpe.sif file can have any name (e.g.,
5 "ristndrd.sif") as long as the extension is .sif. The contents of this file is as
6 follows. Note that the [SetupData] section is essentially the same as in the case in
7 which the image is booted from local disk, with a change to the /rdpath switch.

```
9       [SetupData]
10       BootDevice = "ramdisk(0)"
11       BootPath = "\\<platform>\system32\"
12       OsLoadOptions = "/noguiboot /fastdetect /minint
13       /rdpath=%INSTALLPATH%\%MACHINETYPE%\<bootimage>
14       [/rdimageoffset=<offset>] [/rdimagelength=<length>] [/rdexportascd]
15
16       [RemoteInstall]
17       Repartition = No
18
19       [OSChooser]
20       Description = "<brieif description>"
21       Help = "<longer description>"
22       LaunchFile =
23       "%INSTALLPATH%\%MACHINETYPE%\templates\startrom.com"
24       ImageType = Flat
25       Version = "5.2 (0)"
```

20 [0036] In the above example, the "Repartition = No" line can be used to
21 avoid a warning about the disk being erased that may arise if the Client Installation
22 Wizard is used. The Description and Help text are user selectable. LaunchFile and
23 ImageType should not be changed from that shown above. The Version line is
24 optional.

1 [0037] An example implementation for booting from an optical media
2 image obtained from a non-RIS network boot server is illustrated next.
3 Implementation details specific to RAM disk are illustrated here. The initial boot
4 program is startrom.com. Two files, ntldr (renamed from setupldr.exe) and
5 ntdetect.com are stored in the same directory as startrom.com.

6 [0038] Fig. 3 illustrates another implementation of a booting operation flow
7 or algorithm 300, for booting an optical media image obtained from physical
8 optical media, such as a compact disc (CD) or DVD. The operation flow 300 can
9 be carried out by a single computer. When the computer initially powers-up, the
10 booting process begins from the basic input/output system (BIOS). The BIOS
11 jumps to the boot sector, which loads a boot loader into RAM.
12

13 [0039] A locating operation 302 locates the optical media content that
14 contains the operating system (OS) software on the physical optical media. One
15 implementation of the locating operation 302 determines the location of the OS
16 from an information file that includes the location of the OS, drivers to be loaded,
17 and other information. A copying operation 304 copies the optical media content
18 including the OS into an optical media image in RAM.
19

20 [0040] An implementation of the operation flow 300 employs one or more
21 OS load options for locating and booting from the optical media image in RAM.
22 Exemplary options are shown as follows:
23

24 /RDPATH - specifies the name of the file that contains the optical media
25 image.

1 /RDIMAGEOFFSET - specifies the offset within the file of the start of the
2 optical media image. This is used when the file includes a header. If omitted,
the image offset is assumed to be 0.

3 /RDIMAGELENGTH - specifies the length of the optical media image
4 within the file. In other words, the boot image extends from
RDIMAGEOFFSET to RDIMAGEOFFSET+RDIMAGELENGTH-1. If
5 omitted, the image is assumed to extend from RDIMAGEOFFSET to the
6 end of the file.

7 /RDEXPORTASCD - specifies that once the kernel starts, the RAM disk
8 program should export the RAM disk optical media image as a CD-like
9 device. This means that the RAM disk program reports the device as type
10 FILE_DEVICE_CD_ROM instead of FILE_DEVICE_DISK, and that the
RAM disk program reports a CD-compatible drive geometry (e.g., 2048
bytes per sector, etc.). This switch can be specified for ISO/UDF images.

11 [0041] Prior to executing the operation flow 300, the following directory
12 structure should be burned onto the physical optical media:

13 \winnt.sif
14 \<bootimage>
15 \<platform>
16 bootfix.bin
ntdetect.com
17 setupldr.bin

18 The boot image file can be elsewhere on the physical optical media if necessary.

19 The file 'winnt.sif' has the same contents as in the case of booting from local disk.

20 When the physical optical media is burned, the El Torito boot sector should also be
21 included. The boot image file burned onto the physical optical media can be an
22 ISO, UDF, or UDF-bridge image file.

23 [0042] A jumping operation 306 jumps to a specified location in the optical
24 media image. An emulating operation 308 emulates the physical optical media
25

1 using a RAM disk program and optical media file system driver that provide
2 access to the optical media image in RAM. The emulating operation 308 redirects
3 physical optical media accesses to the appropriate location in the optical media
4 image in RAM. Because the physical optical media is being emulated in RAM, the
5 physical optical media storing the OS can be removed and replaced with another
6 physical optical media, if desired.

7 [0043] Fig. 4 illustrates yet another implementation of a booting operation
8 flow or algorithm 400, for booting an optical media image obtained from a hard
9 disk. The operation flow 400 can be carried out by a single computer. As
10 discussed above, when the computer initially powers-up, the booting process
11 begins from the basic input/output system (BIOS). The BIOS jumps to the boot
12 sector, which loads a boot loader into RAM.

13
14 [0044] A locating operation 402 locates the optical media content that
15 contains the operating system (OS) software on a hard disk. One implementation
16 of the locating operation 402 determines the location of the OS from an
17 information file that includes the location of the OS, drivers to be loaded, and
18 other information. A copying operation 404 copies the optical media content
19 including the OS into an optical media image in RAM.
20

21 [0045] An example of data and steps performed by the locating and copying
22 operation 404 are shown below for booting a WINDOWS PE image copied from a
23 local hard disk. First, copy the boot image (e.g., ISO, UDF or RAM disk) to the
24
25

1 local hard disk. Typically the boot image would be stored at the root of the boot
2 partition, but this is not required. Second, create a text file in the root of the boot
3 partition called winnt.sif. This file should contain the following lines:

```
4 [SetupData]
5 BootDevice = "ramdisk(0)"
6 BootPath = "\"<platform>\system32\"
7 OsLoadOptions = "/noguiboot /fastdetect /minint /rdpath=<bootimage>
8 [/rdimageoffset=<offset>] [/rdimagelength=<length>] [/rdexportascd]
```

8 <platform> is, for example, i386

9 <bootimage> is the path to the boot image file. If the file is on the boot
10 partition, this can be a relative path. For example, if the file is at the root of
11 the boot partition, <bootimage> could be "winpe.iso". If it's in a
12 subdirectory, it could be "tmp\winpe.iso". You can put the file in any ARC-
13 accessible location by specifying a full ARC path, e.g.,
14 "multi(0)disk(0)rdisk(0)partition(2)\winpe.iso".

14 [0046] The file winnt.sif has the same contents as in the implementation in
15 which the computer boots from a local disk. The file winnt.sif is stored in the
16 same directory as the image loader. The boot image file can be in a subdirectory
17 below winnt.sif, if desired.

18 [0047] In the particular implementation shown above, the /rdexportascd
19 must be specified for optical media images, but should be omitted for non-optical
20 RAM disk images.

21 [0048] A jumping operation 406 jumps to a specified location in the optical
22 media image. An emulating operation 408 emulates the hard disk using a RAM
23 disk program and an optical media file system driver that provide access to the
24
25

1 optical media image in RAM. The emulating operation 408 redirects accesses to
2 the hard disk to the appropriate location in the optical media image in RAM.

3 [0049] Because the physical hard disk is being emulated in RAM, the
4 physical hard disk storing the OS can be reformatted or otherwise used to store
5 other data, if desired. If the hard disk is reformatted or the OS is written over on
6 the hard disk, a copying operation 410 copies the optical media image from RAM
7 onto the hard disk before the computer is powered down.
8
9
10

11 **Exemplary Computing Device**

12 [0050] Fig. 5 is a schematic illustration of an exemplary computing device
13 500 that can be utilized to implement a client, server, or other network node.
14 Computing device 500 includes one or more processors or processing units 532, a
15 system memory 534, and a bus, 536 that couples various system components
16 including the system memory 534 to processors 532. The bus 536 represents one or
17 more of any of several types of bus structures, including a memory bus or memory
18 controller, a peripheral bus, an accelerated graphics port, and a processor or local
19 bus using any of a variety of bus architectures. The system memory 534 includes
20 read only memory (ROM) 538 and random access memory (RAM) 540. A basic
21 input/output system (BIOS) 542, containing the basic routines that help to transfer
22
23
24
25

1 information between elements within computing device 500, such as during start-
2 up, is stored in ROM 538.

3 [0051] Computing device 500 further includes a hard disk drive 544 for
4 reading from and writing to a hard disk (not shown), and may include a magnetic
5 disk drive 546 for reading from and writing to a removable magnetic disk 548, and
6 an optical disk drive 550 for reading from or writing to a removable optical disk
7 552 such as a CD ROM, DVD, or other optical media. The hard disk drive 544,
8 magnetic disk drive 546, and optical disk drive 550 are connected to the bus 536
9 by appropriate interfaces 554a, 554b, and 554c. The drives and their associated
10 computer-readable media provide nonvolatile storage of computer-readable
11 instructions, data structures, program modules and other data for computing device
12 500. Although the exemplary environment described herein employs a hard disk, a
13 removable magnetic disk 548 and a removable optical disk 552, other types of
14 computer-readable media such as magnetic cassettes, flash memory cards, digital
15 video disks, random access memories (RAMs), read only memories (ROMs), and
16 the like, may also be used in the exemplary operating environment.

17 [0052] A number of program modules may be stored on the hard disk 544,
18 magnetic disk 548, optical disk 552, ROM 538, or RAM 540, including an
19 operating system 558, one or more application programs 560, other program
20 modules 562, and program data 564. A user may enter commands and information
21 into computing device 500 through input devices such as a keyboard 566 and a
22
23
24
25

1 pointing device 568. Other input devices (not shown) may include a microphone,
2 joystick, game pad, satellite dish, scanner, or the like. These and other input
3 devices are connected to the processing unit 532 through an interface 556 that is
4 coupled to the bus 536. A monitor 572 or other type of display device is also
5 connected to the bus 536 via an interface, such as a video adapter 574.

6 [0053] Generally, the data processors of computing device 500 are
7 programmed by means of instructions stored at different times in the various
8 computer-readable storage media of the computer. Programs and operating systems
9 may be distributed, for example, on floppy disks, CD-ROMs, or electronically, and
10 are installed or loaded into the secondary memory of the computing device 500. At
11 execution, the programs are loaded at least partially into the computing device's
12 500 primary electronic memory.
13

14 [0054] Computing device 500 may operate in a networked environment
15 using logical connections to one or more remote computers, such as a remote
16 computer 576. The remote computer 576 may be a personal computer, a server, a
17 router, a network PC, a peer device or other common network node, and typically
18 includes many or all of the elements described above relative to computing device
19 500. The logical connections depicted in Fig. 5 include a LAN 580 and a WAN
20 582. The logical connections may be wired, wireless, or any combination thereof.
21
22

23 [0055] The WAN 582 can include a number of networks and subnetworks
24 through which data can be routed from the computing device 500 and the remote
25

1 computer 576, and vice versa. The WAN 582 can include any number of nodes
2 (e.g., DNS servers, routers, etc.) by which messages are directed to the proper
3 destination node.

4 [0056] When used in a LAN networking environment, computing device
5 500 is connected to the local network 580 through a network interface or adapter
6 584. When used in a WAN networking environment, computing device 500
7 typically includes a modem 586 or other means for establishing communications
8 over the wide area network 582, such as the Internet. The modem 586, which may
9 be internal or external, is connected to the bus 536 via a serial port interface 556.
10

11 [0057] In a networked environment, program modules depicted relative to
12 the computing device 500, or portions thereof, may be stored in the remote
13 memory storage device. It will be appreciated that the network connections shown
14 are exemplary and other means of establishing a communications link between the
15 computers may be used.
16

17 [0058] The computing device 500 may be implemented as a server computer
18 that is dedicated to server applications or that also runs other applications.
19 Alternatively, the computing device 500 may be embodied in, by way of
20 illustration, a stand-alone personal desktop or laptop computer (PCs), workstation,
21 personal digital assistant (PDA), or electronic appliance, to name only a few.
22

23 [0059] Various modules and techniques may be described herein in the
24 general context of computer-executable instructions, such as program modules,
25

1 executed by one or more computers or other devices. Generally, program modules
2 include routines, programs, objects, components, data structures, etc. that perform
3 particular tasks or implement particular abstract data types. Typically, the
4 functionality of the program modules may be combined or distributed as desired in
5 various embodiments.

6 [0060] An implementation of these modules and techniques may be stored
7 on or transmitted across some form of computer-readable media. Computer-
8 readable media can be any available media that can be accessed by a computer. By
9 way of example, and not limitation, computer-readable media may comprise
10 “computer storage media” and “communications media.”

12 [0061] “Computer storage media” includes volatile and non-volatile,
13 removable and non-removable media implemented in any method or technology
14 for storage of information such as computer-readable instructions, data structures,
15 program modules, or other data. Computer storage media includes, but is not
16 limited to, RAM, ROM, EEPROM, flash memory or other memory technology,
17 CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic
18 cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices,
19 or any other medium which can be used to store the desired information and which
20 can be accessed by a computer.

23 [0062] “Communication media” typically embodies computer-readable
24 instructions, data structures, program modules, or other data in a modulated data
25

1 signal, such as carrier wave or other transport mechanism. Communication media
2 also includes any information delivery media. The term "modulated data signal"
3 means a signal that has one or more of its characteristics set or changed in such a
4 manner as to encode information in the signal. By way of example, and not
5 limitation, communication media includes wired media such as a wired network or
6 direct-wired connection, and wireless media such as acoustic, RF, infrared, and
7 other wireless media. Combinations of any of the above are also included within
8 the scope of computer-readable media.
9

10 [0063] In addition to the specific implementations explicitly set forth
11 herein, other aspects and implementations will be apparent to those skilled in the
12 art from consideration of the specification disclosed herein. It is intended that the
13 specification and illustrated implementations be considered as examples only, with
14 a true scope and spirit of the following claims.
15
16
17
18
19
20
21
22
23
24
25